

SISTEM PEMADAM KEBAKARAN BERBASIS ANDROID



**Disusun sebagai salah satu syarat menyelesaikan Program Studi Strata I pada Jurusan
Informatika Fakultas Komunikasi dan Informatika**

Oleh:

YUDHA ADI PERMANA

L 200 110 147

**PROGRAM STUDI INFORMATIKA
FAKULTAS KOMUNIKASI DAN INFORMATIKA
UNIVERSITAS MUHAMMADIYAH SURAKARTA
2016**

HALAMAN PERSETUJUAN

SISTEM PEMADAM KEBAKARAN BERBASIS ANDROID

PUBLIKASI ILMIAH

oleh:

YUDHA ADI PERMANA

L 200 110 147

Telah diperiksa dan disetujui untuk diuji oleh:

Dosen Pembimbing



Dr. Heru Supriyono, M.Sc.

NIK.970

HALAMAN PENGESAHAN

**JUDUL NASKAH PUBLIKASI ILMIAH MAHASISWA
UNIVERSITAS MUHAMMADIYAH SURAKARTA**

OLEH

YUDHA ADI PERMANA

L 200 110 147

**Telah dipertahankan di depan Dewan Penguji
Fakultas Komunikasi dan Informatika
Universitas Muhammadiyah Surakarta
Pada hari Sabtu, 16 April 2016
dan dinyatakan telah memenuhi syarat**

Dewan Penguji:

- 1. Dr. Heru Supriyono, M.Sc.**
(Ketua Dewan Penguji)
- 2. Dr. Ir. Bana Handaga, M.T.**
(Anggota I Dewan Penguji)
- 3. Hernawan Sulistyanto, S.T., M.T.**
(Anggota II Dewan Penguji)

(.....)
(.....)
(.....)

Publikasi ilmiah ini telah diterima sebagai salah satu persyaratan

Untuk memperoleh gelar sarjana

Tanggal .3...Mei...2016

Mengetahui,

Dekan
Fakultas Komunikasi dan Informatika

Husei Thamrin, S.T., M.T., Ph.D.
NIK : 706

Ketua Program Studi
Informatika

Dr. Heru Supriyono, M.Sc.
NIK:970



**UNIVERSITAS MUHAMMADIYAH SURAKARTA
FAKULTAS KOMUNIKASI DAN INFORMATIKA
PROGRAM STUDI INFORMATIKA**

Jl. A Yani Tromol Pos 1 Pabelan Kartasura Telp. (0271)717417, 719483 Fax (0271) 714448
Surakarta 57102 Indonesia. Web: <http://informatika.ums.ac.id>. Email: informatika@ums.ac.id

SURAT KETERANGAN LULUS PLAGIASI

012/A.3-IL.3/INF-FKI/IV/2016

Assalamu'alaikum Wr. Wb

Biro Skripsi Program Studi Informatika menerangkan bahwa :

Nama : YUDHA ADI PERMANA
NIM : L200110147
Judul : SISTEM PEMADAM KEBAKARAN BERBASIS ANDROID

Program Studi : Informatika
Status : **Lulus**

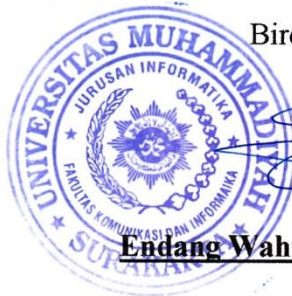
Adalah benar-benar sudah lulus pengecekan plagiasi dari Naskah Publikasi Skripsi, dengan menggunakan aplikasi Turnitin.

Demikian surat keterangan ini dibuat agar dipergunakan sebagaimana mestinya.

Wassalamu'alaikum Wr. Wb

Surakarta, 25 April 2016

Biro Skripsi Informatika



Endang Wahyu Pamungkas, S.Kom., M.Kom.

preferences

previous paper next paper



Processed on: 02-May-2016 08:15 WIB
ID: 668278308
Word Count: 4050
Submitted: 1

SISTEM PEMADAM KEBAKARAN BERBASIS ANDROID

By Yudha Adi Permana

Similarity Index	Similarity by Source
15%	Internet Sources: 13%
	Publications: 2%
	Student Papers: 9%

Document Viewer

exclude quoted exclude bibliography exclude small matches

mode: show highest matches together

SISTEM PEMADAM KEBAKARAN BERBASIS ANDROID Abstrak Sistem deteksi dini pemadam kebakaran otomatis dengan pengendali aplikasi Android merupakan suatu sistem yang dirancang untuk mengatasi permasalahan kebakaran yang sering terjadi pada bangunan dan hal tersebut tergolong masih langka namun keberadaannya diperlukan. Sistem ini dapat mendeteksi terjadinya tanda-tanda kebakaran dan mengirimkan perintah sebagai output berupa penyemprotan air sehingga dapat meminimalisir terjadinya kebakaran. Berdasarkan permasalahan tersebut penelitian ini termasuk jenis penelitian lapangan yaitu mengacu pada penyelesaian suatu masalah yang terjadi di lapangan. Prototype web based online ini terdiri dari aplikasi Android sebagai main program, web server dan rangkain Arduino yang terdiri dari sistem pengendalian dan pengontrolan keamanan. Aplikasi Android terhubung secara serial dengan mikrokontroler pada Arduino yang dimanfaatkan sebagai pengendali otomatis. Sistem dengan perangkat tersebut kemudian mampu mendeteksi adanya tanda-tanda kebakaran dan dapat menyemprotkan air melalui solenoid valve. Hasil uji yang dilakukan pada miniatur rumah pada area A diberi perlakuan pemberian asap korek api, ternyata sensor yang terdapat pada sistem mendeteksi adanya tanda-tanda kebakaran. Kemudian Arduino mengirim pemberitahuan melalui web server, database pada web server bertambah lalu mengirimkan ke GCM (Google Cloud Messenger) dan menyampaikan ke aplikasi Android. Kemudian pengguna melakukan perintah untuk menangani area A dengan menekan tombol on pada area bersangkutan, lalu mengirimkan ke web server dan selanjutnya ke Arduino sehingga bersamaan dengan menerima perintah tersebut solenoid valve terbuka dan air keluar untuk mengatasi area tersebut. Area A dapat ditangani dan Arduino mengirim notifikasi ke aplikasi Android. Hasil pengujian menunjukan bahwa sistem deteksi dini pemadam kebakaran otomatis dapat dikendalikan melalui jarak jauh dengan menggunakan aplikasi Android. Kata Kunci: Deteksi kebakaran, Arduino, Web Server, Aplikasi Android Abstract Early detection system with automatic fire control Android application is a system designed to overcome the problems of fires that often occur in the building and it is still relatively rare, but its existence is necessary. The system can detect the signs fire and send commands as output in the form of spraying water to minimize the fire.

1 2% match (Internet from 04-Jun-2015)
<http://jurnal.upi.edu>

2 2% match (Internet from 14-Oct-2014)
<http://rdi.or.id>

3 2% match (Internet from 22-Mar-2015)
<http://forum.bildr.org>

4 1% match (Internet from 18-Mar-2016)
<http://eprints.undip.ac.id>

5 1% match (Internet from 02-Mar-2015)
<http://blog.revivalx.com>

6 1% match (Internet from 25-Mar-2016)
<http://androidexample.com>

7 1% match (Internet from 20-Apr-2016)
<http://arduinotronics.blogspot.com>

8 1% match (student papers from 21-Jul-2014)

PERNYATAAN

Dengan ini saya menyatakan bahwa dalam skripsi ini tidak terdapat karya yang pernah diajukan untuk memperoleh gelar kesarjanaan di suatu perguruan tinggi dan sepanjang pengetahuan saya juga tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan orang lain, kecuali secara tertulis diacu dalam naskah dan disebutkan dalam daftar pustaka.

Apabila kelak terbukti ada ketidakbenaran dalam pernyataan saya di atas, maka akan saya pertanggungjawabkan sepenuhnya.

Surakarta, 9 Maret 2016

Penulis


YUDHA ADI PERMANA
L 200 110 147

SISTEM PEMADAM KEBAKARAN BERBASIS ANDROID

Abstrak

Sistem deteksi dini pemadam kebakaran otomatis dengan pengendali aplikasi Android merupakan suatu sistem yang dirancang untuk mengatasi permasalahan kebakaran yang sering terjadi pada bangunan dan hal tersebut tergolong masih langka namun keberadaannya diperlukan. Sistem ini dapat mendeteksi terjadinya tanda-tanda kebakaran dan mengirimkan perintah sebagai output berupa penyemprotan air sehingga dapat meminimalisir terjadinya kebakaran. Berdasarkan permasalahan tersebut penelitian ini termasuk jenis penelitian lapangan yaitu mengacu pada penyelesaian suatu masalah yang terjadi di lapangan. *Prototype web based online* ini terdiri dari aplikasi Android sebagai *main program*, *web server* dan rangkaian Arduino yang terdiri dari sistem pengendalian dan pengontrolan keamanan. Aplikasi Android terhubung secara serial dengan mikrokontroler pada Arduino yang dimanfaatkan sebagai pengendali otomatis. Sistem dengan perangkat tersebut kemudian mampu mendeteksi adanya tanda-tanda kebakaran dan dapat menyemprotkan air melalui *solenoid valve*. Hasil uji yang dilakukan pada miniatur rumah pada area A diberi perlakuan pemberian asap korek api, ternyata sensor yang terdapat pada sistem mendeteksi adanya tanda-tanda kebakaran. Kemudian Arduino mengirim pemberitahuan melalui *web server*, database pada *web server* bertambah lalu mengirimkan ke GCM (*Google Cloud Messenger*) dan menyampaikan ke aplikasi Android. Kemudian pengguna melakukan perintah untuk menangani area A dengan menekan tombol on pada area bersangkutan, lalu mengirimkan ke *web server* dan selanjutnya ke Arduino sehingga bersamaan dengan penerimaan perintah tersebut *solenoid valve* terbuka dan air keluar untuk mengatasi area tersebut. Area A dapat ditangani dan Arduino mengirim notifikasi ke aplikasi Android. Hasil pengujian menunjukkan bahwa sistem deteksi dini pemadam kebakaran otomatis dapat dikendalikan melalui jarak jauh dengan menggunakan aplikasi Android.

Kata Kunci: Deteksi kebakaran, Arduino, Web Server, Aplikasi Android

Abstract

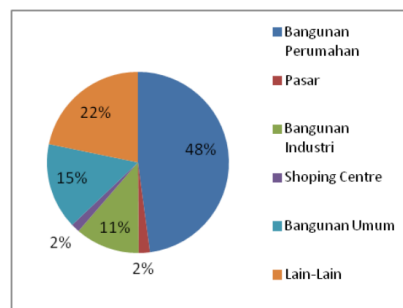
Early detection system with automatic fire control Android application is a system designed to overcome the problems of fires that often occur in the building and it is still relatively rare, but its existence is necessary. The system can detect the signs fire and send commands as output in the form of spraying water to minimize the fire. Based on the problems of this study include the some kind of field research that relies on problem solving that occurs in the field. *Prototype web based online* consists of Android as the main program, a web server and a string of Arduino that costist of systems and security control. Android Apps connected in series with the Arduino microcontroller which is used as automatic controller. The Systems then, is able to detect any signs of fire and can sprinkle the water through a solenoid valve. The results of the tests conducted on the miniature house on the treated area A smoke Award lighters, it turns out that there are sensors on the system detects signs of fire. the Arduino then sends a notification through the web server, the database on the web server then sends to increase GCM (*Google Cloud Messenger*) and delivered to the Android application. Then the user performs a command to handle the area A by pressing on the button on the concerned area and sent to the web server and then to the Arduino so that together with the received command, the solenoid valve will be opened and the water sprikle to extinguish the fire in these

areas. after the Area A has been take care of, the Arduino then, sends notifications to Android. The test result showed that predetected fire fighting system can be remote controlled by Android application.

Keywords: Fire detection, Arduino, Web Server, Android Application.

1. PENDAHULUAN

Sistem keamanan sangat dibutuhkan dalam sebuah bangunan maupun perumahan guna melindungi dari hal-hal yang tidak diinginkan. Untuk memunculkan rasa aman diperlukan alat untuk melindungi rumah dari berbagai ancaman terutama kebakaran yang sering terjadi pada pemukiman penduduk, gudang, dan bangunan lainnya. Sistem keamanan yang baik adalah sistem keamanan yang dapat bergerak sesuai keinginan pemilik, efisien dan tidak merusak sekitar. Dengan adanya mikrokontroler sebagai otak dan teknologi *smartphone*, maka sistem keamanan dapat dibuat menjadi sistem keamanan yang baik. Menurut Huang (2009), menyatakan bahwa salah satu kejadian kebakaran yang paling merugikan adalah kejadian kebakaran di daerah perkotaan permukiman. Berdasarkan data (Dinas Pencegahan dan Penanggulangan Kebakaran Kota Bandung) tahun 2000-2010, terjadi sebanyak 1.624 kebakaran dengan sekitar 773 (48%) kejadian terjadi di daerah perumahan.



Gambar 1. Kejadian Kebakaran Kota Bandung Tahun 2000-2010

Sumber: Dinas Pemadam Kebakaran Kota Bandung, 2010

Berdasarkan uraian masalah diatas, menjadi penting untuk membuat sistem keamanan kebakaran yang baik dan sesuai keinginan dengan mikrokontroler berbasis Arduino Uno. Guntoro, dkk (2013), sistem keamanan dapat dilakukan dengan menggunakan alat elektronik sebagai pengganti sistem keamanan kunci konvensional. Pada pengujian alat keamanan pintu menggunakan solenoid dan mengendalikannya melalui keypad. Alat ini dirancang dengan memanfaatkan mikrokontroler yang berbasis Arduino Uno sebagai pengendali utama, dimana keypad berfungsi sebagai alat input kode password dan memberikan perintah pada mikrokontroler untuk mengendalikan relay. Arduino mikrokontroler dapat mendeteksi input keypad dengan baik, delay yang diterapkan untuk mengaktifkan solenoid dapat berjalan dengan baik, dan fitur untuk mengubah dan menyimpan kode password baru dapat berjalan dengan baik.

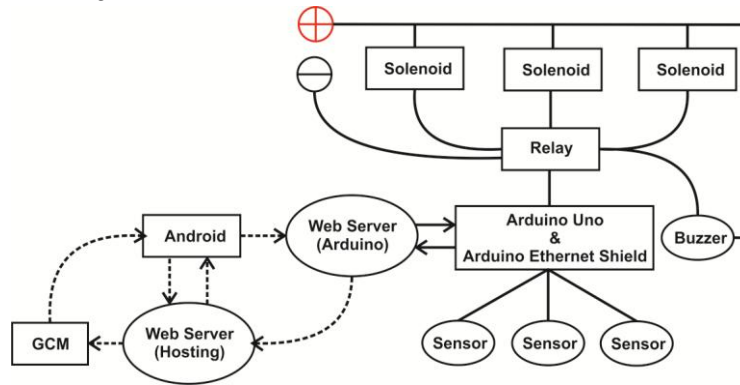
Berkaitan dengan pernyataan tersebut, maka mikrokontroler dapat digunakan sebagai pengendali pendeteksi sistem pemadam kebakaran. Arifiyanto (2013), menyatakan bahwa mikrokontroler dapat dikendalikan melalui jarak jauh dengan bantuan komputer sebagai server dan menggunakan aplikasi untuk komunikasi serial. Data perintah disimpan dalam basis data yang kemudian diambil oleh aplikasi pada komputer server, lalu komputer server mengirimkan perintah pada mikrokontroler sesuai dengan data pada basis data. Kemudian Nur (2010), menyatakan bahwa pendeteksi posisi dengan menggunakan mikrokontroler sangat berguna bagi berbagai penerapan pengendalian. Hal ini dibuktikan dengan melakukan pengujian pengiriman perintah menggunakan sinyal GSM yang ada pada mobile phone lalu dibaca oleh penerima. Kemudian dengan menggunakan rangkaian mikrokontroler yang diintegrasikan dengan pengguna, dimana terdapat memori dan rangkaian pemacu yang mengeksekusi SMS diterima, selanjutnya melakukan pembacaan dan pengiriman perintah. Hal tersebut menunjukkan bahwa mikrokontroler dapat berperan terhadap rumah pintar. Kaur (2010), said that home outomation system can controlled by through smart a chip called microcontrollers. It make automated home can be a very simple grouping of controls, or it can be heavily outomated where any appliance that is pluggd into electrical power is remotely controlled.

Supaya pendeteksi dini sistem pemadam kebakaran bekerja dengan maka optimal perlu dilengkapi dengan komponen-komponen penukung lainnya, seperti sensor asap sebagai pendeteksi adanya tanda kebakaran. Widodo (2003), menyatakan bahwa detektor asap yang digunakan pada alat pendeteksi kebakaran sangat efektif. Dibuktikan pada suatu pengujian dengan sampel transparansi sebagai pengganti asap. Komponen utama alat ini adalah pengindra yang akan mengubah besaran fisis berkas cahaya ke sinyal tegangan analog, transistor sebagai saklar, tiristor sebagai memori (latch), multivibrator yang akan membangkitkan pulsa dan pengeras suara yang menghasilkan bunyi sebagai keluarannya. Hasil pengujian sensor, transistor, tiristor, multivibrator dan pengeras suara menunjukkan alat pendeteksi kebakaran dengan detektor asap dapat bekerja dengan baik. Berdasarkan uraian tersebut maka perlu dibuat alat kewanaman pendeteksi dini sistem pemadam kebakarn dengan komopen-komponen pendukung.

2. METODE

Tugas akhir yang dipublikasikan ini termasuk jenis penelitian terapan yaitu suatu penelitian yang bertujuan untuk memberikan solusi atas permasalahan tertentu secara praktis. Metode yang digunakan adalah metode eksperimen. Waktu yang digunakan untuk menyelesaikan penelitian ini sekitar 12 bulan yaitu bulan Maret 2015 sampai bulan Februari 2016 yang dilakukan di rumah peneliti.

2.1 Gambaran Sistem Kerja

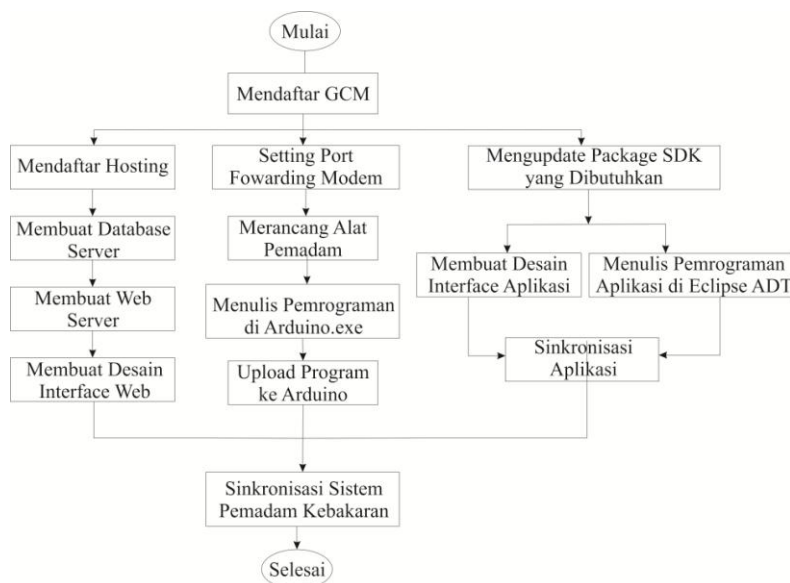


Gambar 2. Gambaran Kerja Sistem Deteksi Dini Pemadam Kebakaran Otomatis

Gambaran sistem kerja ini menunjukkan bagaimana setiap perangkat terhubung. Pada Arduino terdapat sensor yang berfungsi untuk mendeteksi adanya tanda kebakaran, tiga solenoid dan satu buzzer tersambung dengan relay yang terhubung dengan Arduino dan akan merespon ketika kondisi tertentu terpenuhi. Pada Arduino terdapat pula *web server* yang berfungsi sebagai penghubung antara Android dan juga *web server* yang terhubung langsung dengan *database* dan juga GCM (*Google Cloud Messaging*). Android berfungsi sebagai pengendali dan memonitoring sistem jarak jauh, kemudian GCM (*Google Cloud Messaging*) digunakan untuk mengirim pemberitahuan dari Android maupun Arduino secara *real time* melalui data yang masuk *web server*.

2.2 Diagram Alir Pembuatan Sistem

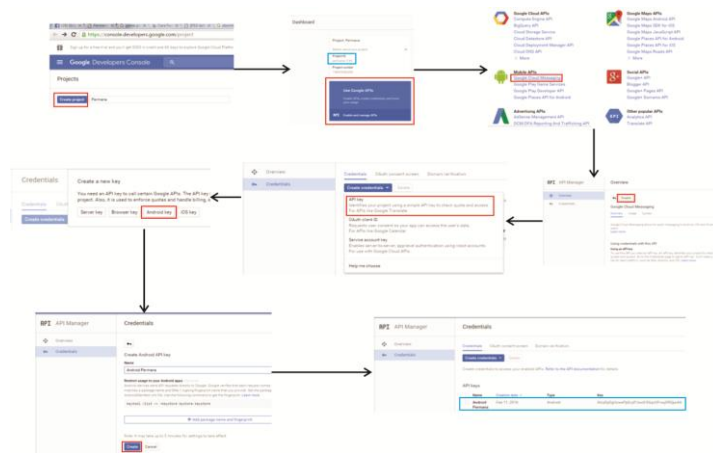
Diagram alir pembuatan sistem pada penelitian ini dapat dilihat pada Gambar 3.



Gambar 3. Diagram Alir Pembuatan Sistem

2.3 Langkah Pembuatan

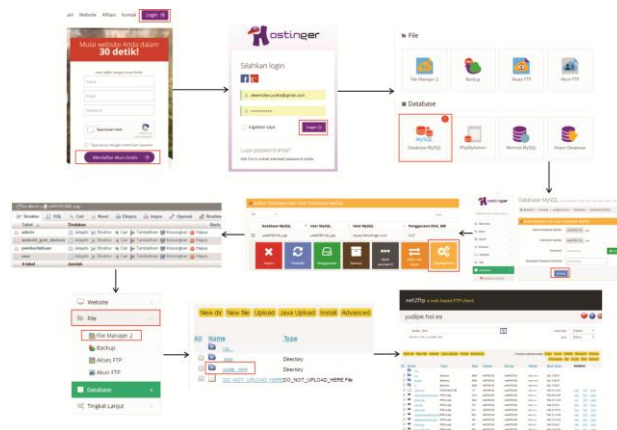
2.3.1 Tahap Pendaftaran Google Cloud Messaging (GCM)



Gambar 4. Tahap Pendaftaran Google Cloud Messaging (GCM)

Masuk ke alamat <http://console.developers.google.com/project> melalui browser untuk mendaftar GCM, kemudian membuat *project* baru dan akan mendapatkan *project number*. Pada penelitian ini *project Id* yang telah dibuat yaitu *Permana-1156* dengan *project number* 154393683280. Selanjutnya masuk ke *Use Google APIs* dan kemudian pilih *Google Cloud Massaging* lalu aktifkan. Setelah itu pilih *Credentials* pada bagian kiri tampilan kemudian pilih buat *Credentials* baru dan pilih *Android key* kemudian isikan *form* yang tersedia. Setelah *Android API key* dibuat maka akan muncul informasi berkaitan dengan *API key* tersebut.

2.3.2 Tahap Pembuatan Web Server



Gambar 5. Tahap Pembuatan Web Server

Untuk membuat *Web Server* yang berfungsi sebagai penghubung dari sistem pemadam kebakaran maka dibutuhkan *browser google chrome* namun sebelum mulai dengan membuat subdomain di www.idhostinger.com. Buatlah database di subdomain yang telah dibuat, kemudian olahlah database tersebut pada *phpMyAdmin*, dan buatlah tabel yang ada di dalam database tersebut. Berikut adalah file *addpemberitahuan.php* untuk mengirim pemberitahuan yang masuk ke GCM:

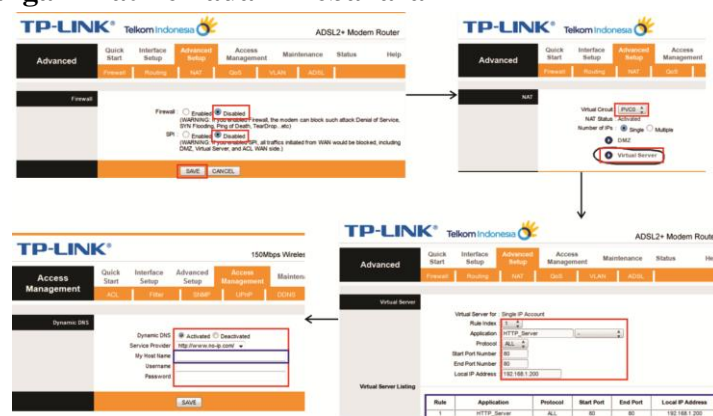
```

...
8| if(isset($_REQUEST['usernamef']) && isset($_REQUEST['pemberitahuan'])) &&
9| isset($_REQUEST['dari'])){
10| $usernamef = $_REQUEST['usernamef'];
11| $pemberitahuan = $_REQUEST['pemberitahuan'];
12| $dari = $_REQUEST['dari'];
...
20| $fields = array(
21|     'registration_ids' => $registrationIDs,
22|     'data'              => array( "message" => $pemberitahuan ),);
23| $headers = array(
24|     'Authorization: key=' . $apiKey,
25|     'Content-Type: application/json');
...
30| $ch = curl_init();
31| curl_setopt( $ch, CURLOPT_URL, 'https://android.googleapis.com/gcm/send');
32| curl_setopt( $ch, CURLOPT_POST, true );
33| curl_setopt( $ch, CURLOPT_HTTPHEADER, $headers);
34| curl_setopt( $ch, CURLOPT_RETURNTRANSFER, true );
35| curl_setopt( $ch, CURLOPT_POSTFIELDS, json_encode( $fields ) );
36| $result = curl_exec($ch);
...

```

Dari pemrograman di atas menjelaskan bahwa baris 8 dan 9 ketika data yang dimaksud masuk maka pemrograman yang ada di dalamnya akan dieksekusi. Kemudian baris 10 sampai 12 adalah inisialisasi dari data yang tadi masuk agar lebih memudahkan untuk diolah nantinya. Pada baris 20 sampai 22 menunjukkan isi dari `$fields` tentang registrasi ID GCM pada aplikasi Android yang dituju dan pesan yang akan ditampilkan pada notifikasi. Selanjutnya adalah baris 23 sampai 25 yang menunjukkan isi dari `$headers` tentang API key GCM dan format pesan yang akan dikirim. Kemudian baris 30 sampai 36 adalah format untuk mengirim data ke alamat GCM seperti `$headers` dan `$fields` dari bentuk array ke bentuk json.

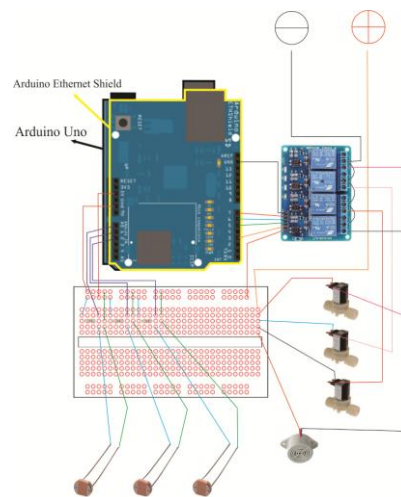
2.3.3 Tahap Perancangan Alat Pemadam Kebakaran



Gambar 6. Langkah Port Forwarding pada Modem TP-Link

Sebelum mulai merancang sistem deteksi pemadam kebakaran, terlebih dahulu mengatur *port forwarding* pada modem, agar dapat mengakses ke *Local IP* yang tersambung dengan modem melalui *Public IP*. Sambungkanlah Laptop dengan TP-Link kemudian buka *browser* dan buka alamat IP

modem TP-Link lalu isi *username* dan *password* kemudian masuk tampilan modem TP-Link dan pilihlah menu *Advanced Setup*. Selanjutnya matikan *firewall* agar akses *port forwarding* diperbolehkan. Kemudian pilih menu *NAT* dan memilih *Virtual Circuit* dimana muncul *Virtual Server* di bawahnya. Isilah form untuk *port forwarding* kemudian informasi yang dimasukan muncul di bawahnya. Dengan ini *port forwarding* selesai dan alamat *local IP* yang telah di masukan akan langsung terakses ketika *public IP* modem diakses pada perangkat lain. Karena alamat *public IP* selalu berganti maka perlu mendaftar ke DDNS terlebih dahulu pada website <http://www.no-ip.com>. Setelah selsesai mendaftar DDNS kemudian bukalah halaman modem TP-Link dan pilih *Access Management* lalu pilih DDNS dan isi *form* yang ada di bawahnya sesuai dengan saat mendaftar di website www.no-ip.com. Setelah *Port Forwarding* berhasil dilakukan peneliti dapat mulai merancang alat pemadam kebakaran.



Gambar 7. Rancangan Alat Pemadam Kebakaran

Dapat dilihat pada Gambar 7, sistem pemadam kebakaran ini menggunakan sensor yang dapat mendeteksi asap. Selain itu juga menggunakan *buzzer* sebagai alarm. Kemudian ada *solenoid valve* yang berfungsi sebagai keran air yang dapat menutup maupun membuka sesuai *relay* yang terhubung dengan Arduino yang berfungsi sebagai *switch* aliran listrik yang masuk menuju *solenoid*. Selanjutnya ada Arduino Ethernet Shield yang terhubung dengan Arduino yang digunakan untuk menghubungkan Arduino dengan modem TP-Link agar dapat terkoneksi dengan internet. Supaya sistem berjalan sesuai dengan kebutuhan maka buatlah program dan *upload* program ke Arduino. Berikut potongan programnya:

```
...
30|void setup() {
...
50|  if (Ethernet.begin(mac) == 0) {}
51|  Ethernet.begin(mac, ip, gateway, gateway, subnet);
...

```

```

62|void loop() {
...
83|   if (client.connect("yudipe.hol.es",80)) {
84|       client.println("POST /addpemberitahuan.php HTTP/1.1");
85|       client.println("Host: yudipe.hol.es");
86|       client.println("Content-Type: application/x-www-form-urlencoded");
87|       client.print("Content-Length: ");
88|       client.println(data.length());
89|       client.println();
90|       client.print(data); }
...
144|EthernetClient client = server.available();
145|   if (client) {
146|       while (client.connected()) {
147|           if (client.available()) {
148|               char c = client.read();
149|               if (readString.length() < 100) {
150|                   readString += c;
151|               }
152|               if (c == '\n') {
153|                   Serial.println(readString);
154|                   client.println("HTTP/1.1 200 OK"); //send new page
155|                   client.println("Content-Type: text/html");
156|                   client.println();
157|                   client.println("<HTML>");
158|                   client.println("<HEAD>");
...
169|                   client.println("<a href='\"/?button1on\"'></a>");
170|                   client.println("<a href='\"/?button1off\"'></a><br />");
171|                   client.println("<a href='\"/?button2on\"'></a>");
172|                   client.println("<a href='\"/?button2off\"'></a><br />");
173|                   client.println("<a href='\"/?button3on\"'></a>");
174|                   client.println("<a href='\"/?button3off\"'></a><br />");
175|                   client.println("</BODY>");
176|                   client.println("</HTML>");
177|                   delay(1);
178|                   client.stop();
179|                   if (readString.indexOf("?button1off") > 0){
180|                       digitalWrite(valve1, HIGH);}
...
200|                   readString="";
...

```

Dari pemrograman di atas menjelaskan bahwa baris 50 sampai 51 berfungsi supaya Arduino dapat dijadikan sebagai *client* dan dapat mengirim data ke *web server* lain dan pada baris 51 berfungsi agar Arduino dapat berjalan sebagai *web server* dan dapat menerima data dari *web server* lain. Kemudian baris 83 menunjukkan Arduino akan tersambung pada *web server* dengan alamat dan *port* tersebut saat sedang menjadi *client*. Selanjutnya baris 84 menunjukkan *post method* yang digunakan dalam mengirim data dan halaman yang tertuju pada *web server* yang tersambung dan baris 85 menunjukkan nama *hosting* atau alamat yang tersambung tadi. Pada baris 86 menunjukkan jenis *content* yang akan dikirim. Kemudian pada baris 87 sampai 89 berfungsi untuk melihat seberapa panjang *content* tersebut dan pada baris 90 memasukan data yang telah terinisialisasi sebelumnya yang akan dikirim ke *web server*. Selanjutnya pada baris 144 berfungsi sebagai

inisialisasi awal supaya Arduino bisa digunakan sebagai *client* dan pada baris 145 jika Arduino dijadikan sebagai *client* maka program di dalamnya akan menginisialisasi agar dapat membaca program dari sisi *client* dan menanggapi permintaan *client* dari baris 146 sampai 152. Pada baris 153 serial monitor akan ikut memantau program dan pada baris 154 akan mengirim halaman baru dengan jenis protocol HTTP. Kemudian pada baris 155 menunjukkan jenis *content* dan pada baris 157 sampai 158 awal dari *web client* lalu pada baris 169 sampai 174 merupakan salah satu isi dari *client* yang berfungsi sebagai link untuk mengirim perintah ke Arduino dari sisi *client*. Selanjutnya pada baris 175 sampai 176 merupakan akhir dari halaman *client* dan pada baris 177 sampai 178 digunakan untuk waktu tunda saat Arduino akan *looping* program dan menghentikan sisi *client*. Pada baris 179 berfungsi sebagai gerbang, ketika saat Arduino berfungsi sebagai *web server* menunjuk link tersebut maka program yang ada di dalamnya akan berjalan seperti pada baris 180 yang akan membuat `valve1` bernilai `HIGH` dan pada baris 200 digunakan untuk membersihkan program sebelumnya agar dapat digunakan dalam program berikutnya.

2.3.4 Tahap Pembuatan Aplikasi Android

Untuk membuat aplikasi Android peneliti menggunakan Aplikasi Eclipse dan Android SDK. Sebelum membuat aplikasi maka update *package Google Cloud Massaging* menggunakan SDK Manager. Setelah melakukan update *Pakage GCM* dan daftar GCM pada situs <http://console.developers.google.com/project>, langkah selanjutnya yaitu bukalah Eclipse. Kemudian buat projek baru di Eclipse dan pilih menu File - New - Android Application lalu akan muncul tampilan *form* untuk membuat *project* baru Android. Jika form *project* baru Android telah selesai diisi langkah selanjutnya yaitu buatlah beberapa *class* dan *layout* baru dan *class* yang harus ada adalah seperti berikut:

2.3.4.1 GCMIntentService.java

```
...
25|private int count = 0;
26|@Override
27|protected void onMessage(Context context, Intent intent) {
28|    String message = intent.getStringExtra("message");
29|    NotificationManager notificationManager = (NotificationManager)
30|    context.getSystemService(Context.NOTIFICATION_SERVICE);
31|    Notification note = new Notification(R.drawable.pc,
32|    getResources().getString(R.string.app_name), System.currentTimeMillis());
33|    Intent notificationIntent = new Intent(context, pemberitahuan.class);
34|    notificationIntent.setFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP |
35|    Intent.FLAG_ACTIVITY_SINGLE_TOP);
36|    PendingIntent pendingIntent = PendingIntent.getActivity(context, 0,
37|    notificationIntent, 0);
38|    note.setLatestEventInfo(context,
39|    getResources().getString(R.string.app_name), message, pendingIntent);
40|    note.number = count++;
41|    note.defaults |= Notification.DEFAULT_SOUND;
42|    note.defaults |= Notification.DEFAULT_VIBRATE;
```



```

43|    note.defaults |= Notification.DEFAULT_LIGHTS;
44|    note.flags |= Notification.FLAG_AUTO_CANCEL;
45|    notificationManager.notify(0, note);}
...

```

Dari pemrograman di atas menjelaskan bahwa baris 25 memberi nilai awal pada `count = 0`. Kemudian pada baris 28 memberi nilai pada `message` dengan data dari GCM yang masuk dengan nama `message`. Selanjutnya pada baris 29 sampai 30 berfungsi supaya mendapatkan penanganan ke `NOTIFICATION_SERVICE` dan pada baris 31 sampai 32 berfungsi untuk mengatur tampilan pada notifikasi yang datang. Kemudian baris 33 akan langsung membuka aplikasi pada *class* yang dimasukan ketika notifikasi di yang muncul disentuh. Pada baris 34 sampai 35 berfungsi agar saat ada notifikasi baru, notifikasi tidak menumpuk tetapi hanya akan menggantikan notifikasi yang lama dan belum hilang dan pada baris 36 sampai 39 berfungsi supaya aplikasi memperbolehkan notifikasi bisa mengakses aplikasi ini sesuai dengan data. Sedangkan baris 40 agar nomor notifikasi sesuai dengan nilai dari `count` yang bertambah dan baris 41 sampai 43 berfungsi untuk mengatur suara, getaran, cahaya layar saat notifikasi muncul. Kemudian pada baris 44 berfungsi untuk menghilangkan notifikasi ketika notifikasi yang muncul disentuh dan baris 45 untuk mengeluarkan notifikasi sesuai dengan pengaturan pada baris sebelumnya.

2.3.4.2 perintah.java

```

...
77|List<NameValuePair> nameValuePairs = new ArrayList<NameValuePair>(3);
78|nameValuePairs.add
79|(new BasicNameValuePair("usernamef",SessionManager.KEY_USERNAME.toString()));
80|nameValuePairs.add
81|(new BasicNameValuePair("pemberitahuan", "area A" + a.toString()));
82|nameValuePairs.add
83|(new BasicNameValuePair("dari", "Android".toString()));
84|try {
85|HttpPost request = new HttpPost("http://yudipe.hol.es/addpemberitahuan.php");
86|request.setEntity(new UrlEncodedFormEntity(nameValuePairs));
87|HttpResponse response = client.execute(request);
88|getRequest("http://" + SessionManager.KEY_IP + "/?buttonlon");
89|} catch (Exception e){}
...

```

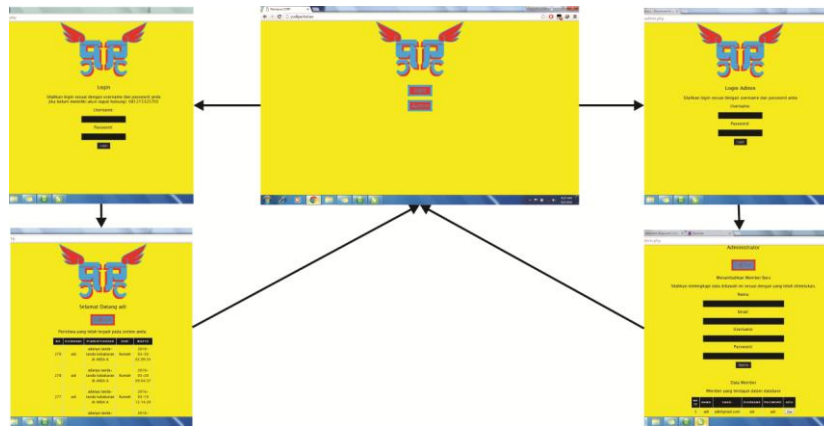
Dari pemrograman di atas menjelaskan bahwa baris 77 menginisialisasi *array list* baru dengan nama `nameValuePairs`. Kemudian pada baris 78 sampai 83 berfungsi menyimpan data yang tersebut pada `nameValuePairs`. Selanjutnya pada baris 85 sampai 87 untuk mengirim data pada `nameValuePairs` menggunakan *post method* yang menunjuk pada alamat *web server* tersebut. Kemudian pada baris 88 berfungsi untuk mengirim perintah ke *Arduino* melalui *web server* pada *Arduino* menggunakan *get method*.

3. HASIL DAN PEMBAHASAN

3.1 Hasil

Penelitian yang dilakukan bertujuan untuk membuat sebuah sistem pemadam kebakaran berbasis mikrokontroler berbasis Android. Pembuatan sistem pemadam kebakaran berbasis arduino ini dilakukan setelah menganalisis kebutuhan, baik kebutuhan *hardware* dan *software*. Hasil pembuatan dan pengujian rangkain sistem pemadam kebakaran berbasis Android adalah sebagai berikut :

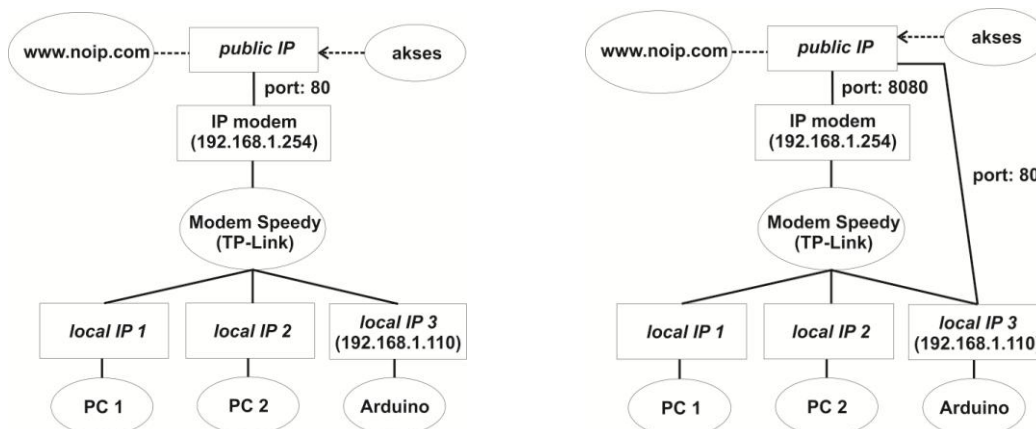
3.1.1 Web Server



Gambar 8. Web Server

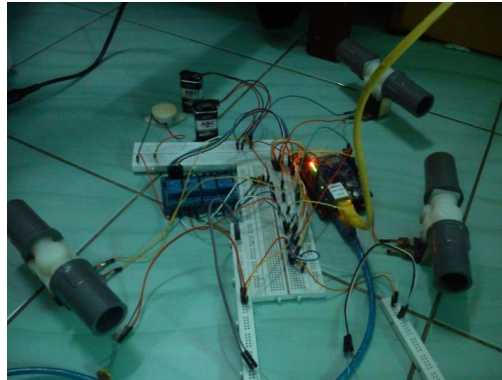
Web server pada sistem pemadam kebakaran dalam penelitian ini dapat diakses menggunakan aplikasi *browser* dengan alamat <http://yudipe.hol.es>. Pada halam utama terdapat dua pilihan tombol yaitu login dan admin. Menu pilihan login merupakan fasilitas untuk menuju *form login user*, setelah memasukan *username* dan *password* yang sesuai maka pengguna dapat masuk ke halaman *user* yang berfungsi hanya untuk melihat pemberitahuan sistem dari user yang *login*. Sedangkan menu pilihan admin merupakan fasilitas untuk menuju *form login admin*, setelah memasukan *username* dan *password* yang sesuai maka admin dapat masuk halam admin yang berfungsi menambah maupun menghapus *user* dan juga melihat identitas para *user*.

3.1.2 Sistem Pemadam Kebakaran



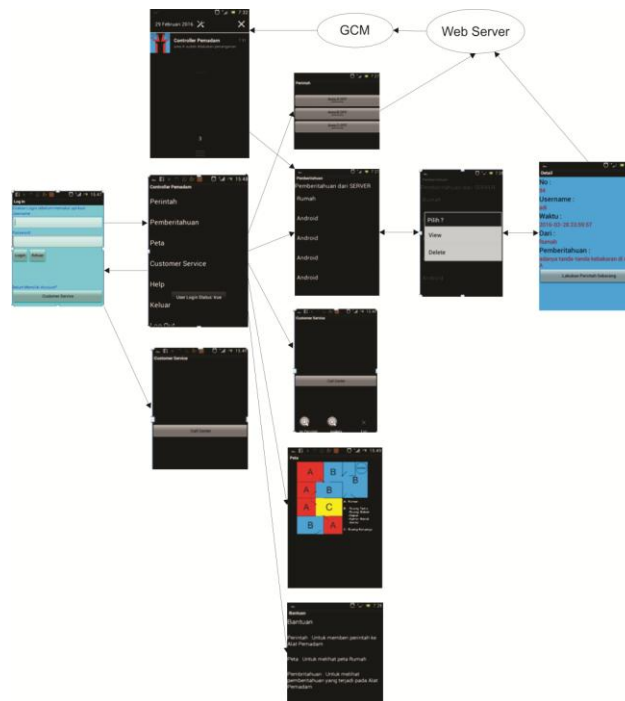
Gambar 9. Sebelum (kiri) dan Sesudah (kanan) Port Forwarding

Pada gambar 9 sebelum *port forwarding* (kiri) dan sesudah *port forwarding* (kanan) dapat dilihat perbedaannya jika sebelum *port forwarding*, *public IP* hanya bisa mengakses *IP* pada modem *speedy* namun setelah *port forwarding* diatur maka dengan mengakses *public ip* bisa terhubung langsung dengan Arduino. Kemudian rangkaian dari berbagai komponen yang dibutuhkan dalam penelitian ini hasilnya dapat dilihat pada gambar 10.



Gambar 10. Rangkaian Alat Pemadam Kebakaran

3.1.3 Aplikasi Android



Gambar 11. Aplikasi Android

Untuk memulai aplikasi saat pertama kali, pengguna harus login terlebih dahulu. Jika belum memiliki *account* pengguna dapat menghubungi admin melalui tombol *Customer Service*. Satu aplikasi hanya dapat digunakan oleh satu *account*. Setelah melakukan login akan muncul notifikasi “*user login status true*”. Pada tampilan utama aplikasi, terdapat beberapa menu pilihan. Untuk melakukan perintah ke Arduino maka pilih menu perintah yang di dalamnya terdapat tiga tombol

perintah. Setelah menekan tombol perintah tersebut maka data pada *database web server* akan bertambah dan *web server* akan mengirim data tersebut ke GCM, kemudian GCM akan mengirim *notifikasi* ke aplikasi Android

Pada menu pemberitahuan terdapat dua macam pemberitahuan berdasarkan asalnya yaitu pemberitahuan berasal dari rumah dan Android. Pemberitahuan dari rumah merupakan pemberitahuan yang berasal dari Arduino yang mendeteksi adanya tanda-tanda kebakaran dan setiap pemberitahuan dapat dilihat detailnya maupun dihapus dengan adanya pilihan saat menekan salah satu pemberitahuan yang ada. Ketika melihat detail pemberitahuan yang berasal dari rumah maka terdapat tombol pada bagian bawah yaitu “lakukan perintah sekarang”, tombol tersebut berfungsi sebagai penanganan langsung tanpa harus ke menu perintah. Kemudian untuk dapat melihat letak area pada rumah dapat menekan tombol “Peta”. Tombol “Customer Service” berfungsi untuk menghubungi admin. Tombol “Help” berfungsi untuk bantuan menggunakan aplikasi. Tombol “Keluar” untuk keluar dari aplikasi. Tombol *log out* untuk keluar dari *account*..

3.2 Pembahasan

3.2.1 Pemberitahuan dari Arduino

Hasil uji coba yang dilakukan sesuai dengan harapan yaitu Arduino dapat mengirim pemberitahuan ke aplikasi Android. Area A diberi asap dengan menggunakan plastik yang terbakar, maka area tersebut menjadi berasap dan asap tersebut menutupi sensor. Pengujian asap lebih baik menggunakan sensor asap tapi bisa digunakan alternatif lain yaitu LDR sebagai sensor cahaya yang akan mendeteksi tingkat kegelapan dari asap yang menutupi LDR. Pengujian untuk mengetahui faktor yang dapat merespon LDR sebagai ganti dari sensor asap dapat dilihat pada tabel berikut:

Tabel 1. Faktor Penguji Respon Sensor

No	Faktor	Tanggapan
1	Uap Air	Tidak Merespon
2	Kabut	Tidak Merespon
3	Asap Putih	Tidak Merespon
4	Asap Hitam	Merespon

Dari data di atas dapat diketahui bahwa hanya asap yang berwarna hitam atau gelap yang dapat ditanggapi LDR karena asap yang berwarna hitam akan menutupi permukaan LDR dari cahaya. Kemudian ketika Arduino merespon adanya tanda-tanda kebakaran, maka Arduino akan mengirim data ke *web server* dan *web server* mengirim ke GCM lalu sampai pada aplikasi Android berupa pemberitahuan bahwa area A terdapat tanda-tanda kebakaran. Data kecepatan penerimaan pemberitahuan dari Arduino ke aplikasi Android dapat dilihat pada tabel berikut:

Tabel 2. Kecepatan Penerimaan Pemberitahuan

Penerimaan Ke-	Kecepatan Pengiriman	
	Cerah	Hujan
1	5,010 detik	4,355 detik
2	3,135 detik	4,110 detik
3	3,522 detik	7,198 detik
4	4,156 detik	3,432 detik
5	3,974 detik	11,967 detik

Tabel tersebut menunjukkan pengaruh cuaca terhadap waktu pengiriman data. Dari lima percobaan tersebut dapat diketahui bahwa cuaca cerah menunjukkan kecepatan pengiriman pemberitahuan yang stabil yaitu antara 3 sampai 5 detik. Sedangkan ketika hujan menunjukkan kecepatan pengiriman pemberitahuan yang tidak stabil karena koneksi internet saat hujan juga tidak stabil.

3.2.2 Pengiriman Perintah dan Penerimaan Pemberitahuan pada Aplikasi Android.

Pengiriman perintah dapat dilakukan ketika aplikasi menerima maupun tidak menerima pemberitahuan dari Arduino. Diketahui pada pengujian pemberitahuan yang dikirim dari Arduino ke aplikasi Android berhasil dengan menunjukkan pemberitahuan bahwa area A terdapat tanda-tanda kebakaran, dan pada detail pemberitahuan di bagian bawah detail pemberitahuan tersebut terdapat tombol “lakukan perintah sekarang”. Untuk melakukan pengujian perintah dari aplikasi Android maka bisa menekan tombol tersebut atau dari tombol yang ada pada menu Perintah. Setelah melakukan perintah muncul pemberitahuan sebagai notifikasi pada aplikasi Android dan Arduino menerima perintah tersebut, kemudian Arduino akan melakukan penanganan dan pada aplikasi Android muncul pemberitahuan bahwa area tersebut sudah dilakukan penanganan. Berikut adalah tabel kecepatan pengiriman perintah dari Android ke Arduino:

Tabel 3. Kecepatan Pengiriman Perintah

Pengiriman Ke-	Kecepatan Pengiriman	
	Cerah	Hujan
1	3,753 detik	4,900 detik
2	3,342 detik	3,118 detik
3	3,124 detik	7,489 detik
4	3,190 detik	8,306 detik
5	5,067 detik	4,817 detik

Tabel tersebut menunjukkan pengaruh cuaca terhadap waktu pengiriman data. Dari lima percobaan tersebut dapat diketahui bahwa cuaca cerah menunjukkan kecepatan pengiriman perintah yang stabil yaitu antara 3 sampai 5 detik. Sedangkan ketika hujan menunjukkan kecepatan pengiriman perintah yang tidak stabil karena koneksi internet saat hujan juga tidak stabil.

4. PENUTUP

Berdasarkan hasil pengujian dan analisis yang dilakukan dapat disimpulkan bahwa sistem deteksi dini pemadam kebakaran otomatis dapat dikendalikan melalui jarak jauh dengan menggunakan aplikasi Android. Arduino dapat mengirim pemberitahuan melalui *web server* sebagai database kemudian ke GCM dan akhirnya diterima oleh aplikasi Android. Perintah dari aplikasi Android dapat diterima oleh Arduino dan diaplikasikan kepada *solenoid valve* untuk mengalirkan air.

DAFTAR PUSTAKA

- Arifiyanto, Farid. 2013. Perancangan *prototype web-based online smart home controlled by smartphone*. Semarang : Universitas Diponegoro. Hal : 7-8.
- Guntoro, Helmi dkk. 2013. *Rancang Bangun Magnetic Door Lock menggunakan Keypad dan Solanoid Berbasis Mikrokontroler Arduino Uno*. Jurnal Electrans. Vol 12, No 1, Hal 39.
- Huang, K., 2009. *Population and Building Factors That Impact Residential Fire Rates in Large U.S. Cities*. Texas :Texas State University-San Marcos.
- Kaur, Inderpreet. 2010. *Microcontroller Based Home Automation System With Security*. *International Journal of Advanced Computer Science and Applications*. Mohali India : Rayat and Bahra Institute of Engineering and Bio-technology. Vol 1, no 6, page 64-65.
- Masianbow, Vidy dkk. 2014. *Pengendali Saklar Listrik Melalui Ponsel Pintar Android*. E-Jurnal Teknik elektro dan Komputer. Vol -, hal 1-9.
- Nur, Anna Nazilah Chamim. 2010. *Penggunaan Microcontroler sebagai Pengendali Posisi dengan Sinyal GSM*. Yogyakarta : Politeknik PPKP. Vol 4, no 1, hal : 438.
- Sagala, Saut dkk. 2013. *Analisis Upaya Pencegahan Bencana Kebakaran di Permukiman* .Jurnal RDI. Vol -, no 3, hal : 6.
- Widodo, Catur Edi. 2003. *Pembuatan Alat Pendeteksi Kebakaran dengan Detektor Asap*. Jurnal Berkala Fisika. Vol 6, No 3, hal : 51.